Progress Report One

Arana Charlebois, Tristan
Carvalho, Cam
Corbett, Cole
Kassie, Turner
Pauls, Andrew
Wallace, Jordan

Cosc 4P02
Dr. Naser Ezzati-Jivan

February 23, 2025

Contents

1 Introduction

As specified in both the project proposal and the release planning document, the goal of our system is to provide a service capable of determining a route for travel based on locations that have been specified by a user. Also, such a system can have value in the logistics and shipping sector, specifically for companies who seek to find an acceptable way to organize their many delivery drivers. We truly believe that this is just the basics for a so called "Logistics System" and that the potential this idea carries is very valuable in numerous areas of logistics.

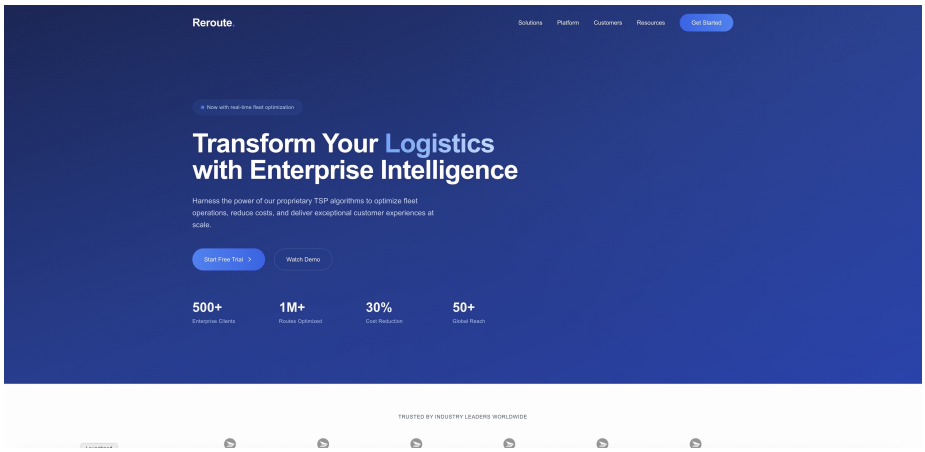
Within the two previous documents pertaining to the development of our Logistics System, we outlined both the functional and non-functional requirements, user stories as well as the tentative timeline of all of our sprints. We also specified the subsystems, such as the genetic algorithm (GA), front-end and user database, along with their interactions within the tasks corresponding to their respective user stories. Within this document, we will provide details about the current state of our system, compare and contrast the actual progression of development with our initial plan and identify challenges faced throughout development. Further, we will give our overall thoughts on the progression of the system and how we plan to not only maintain but improve the system's progression. Finally, we will present the contribution of each member to not only the development of the system, but to the project as a whole as well as the Github activity for all repositories involved in the project.

2 System Details and Current State

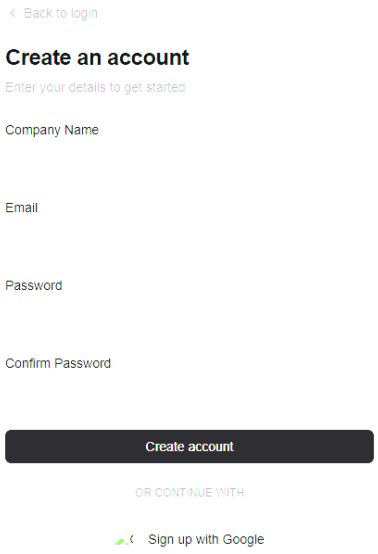
To begin, we would like to showcase the work that the group has completed in terms of the development of a usable system. The system has two main components in development, so this section will be delivered in two parts, namely the work being completed on the front-end of the system and the work completed on the back-end.

2.1 Front-end Development

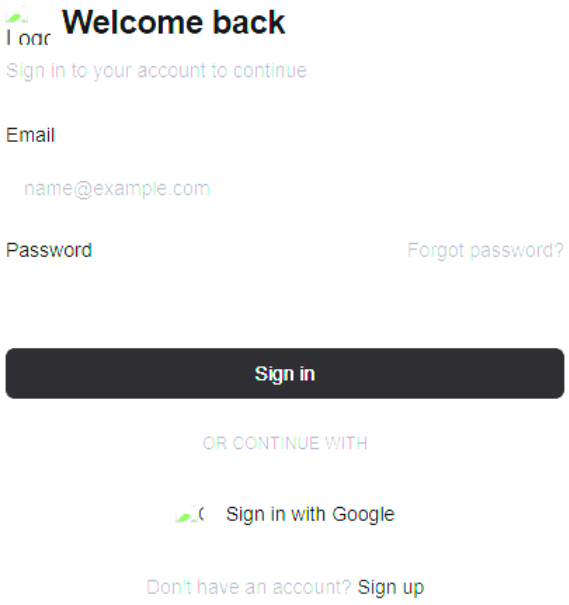
The front-end development is concerned with the completion of a user database, a landing page, and the actual site where users are able to set locations, and additional fields to effect the route produced by the GA. As of right now, a sample of the landing page has been created, and is displayed in the figure below. Refining this landing page is within the next steps of the project, as we plan to make it more consistent with the features that the system can offer reliably, rather than have information based on the early plans of the system.



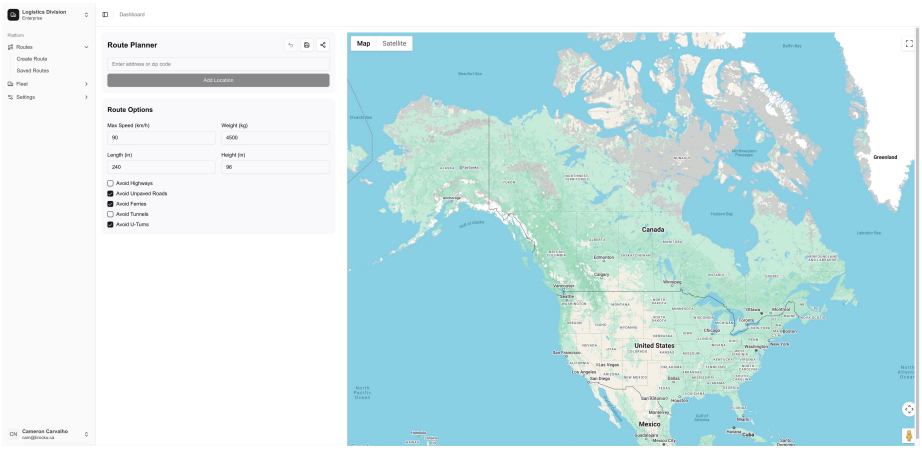
Next, we consider the user database, which currently allows user to create accounts by filling in the respective fields.



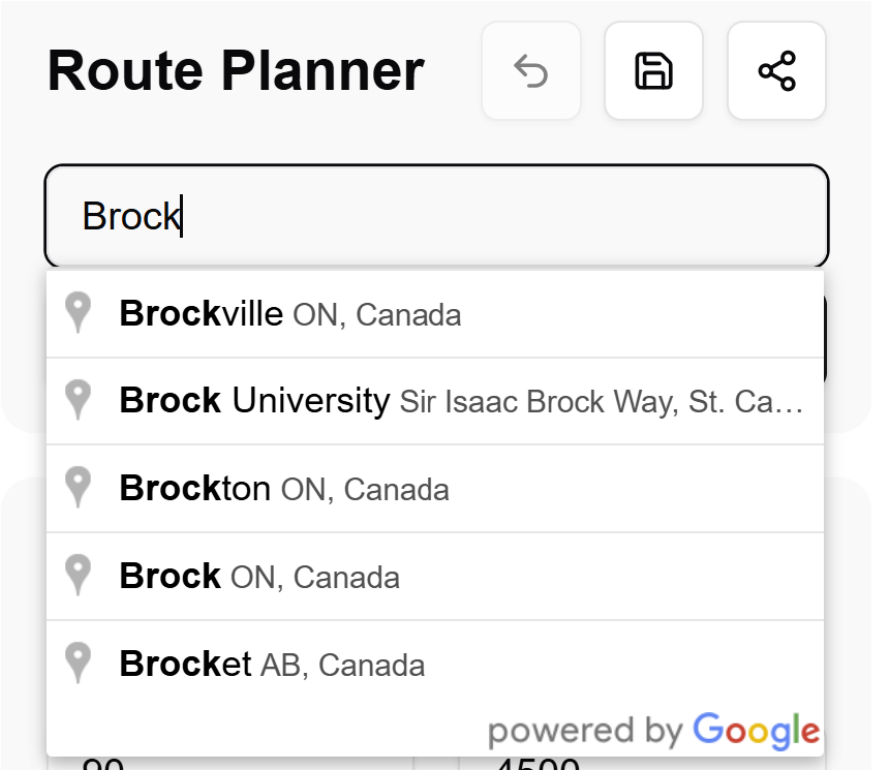
Also, there an account login page allowing users to input their login credentials. Note that following the "Sign Up" link will lead to the create account page shown above.



We examine the sample deployment of the front-end, where users are able to interact with locations, routes and filling out other available fields. The number and sophistication of the fields are a matter of overall project velocity, but the following screenshot communicates the general idea of what to expect.

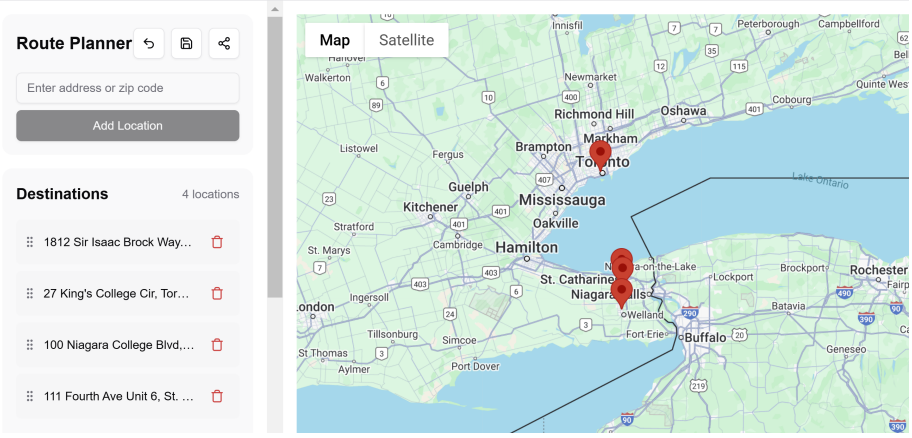


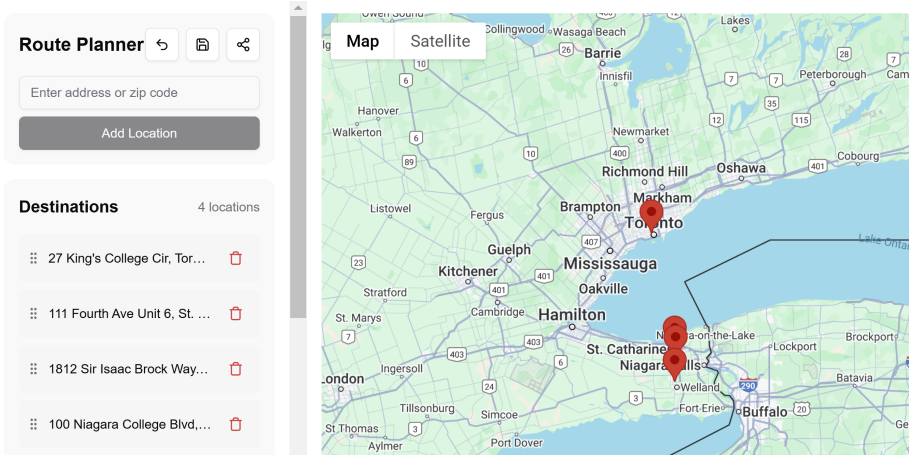
Users can also add locations, for example, Brock University, and searching for the location will auto-complete as shown below.



Locations can then be added into a list and then with the optimize route function, it will pass the information to the GA, reordering the locations based

on the results of the GA. The described behavior can be observed in the following two screen captures, the first being the input of four locations in a randomized order, and the second screen capture showing the reordering of locations based upon the time to complete.



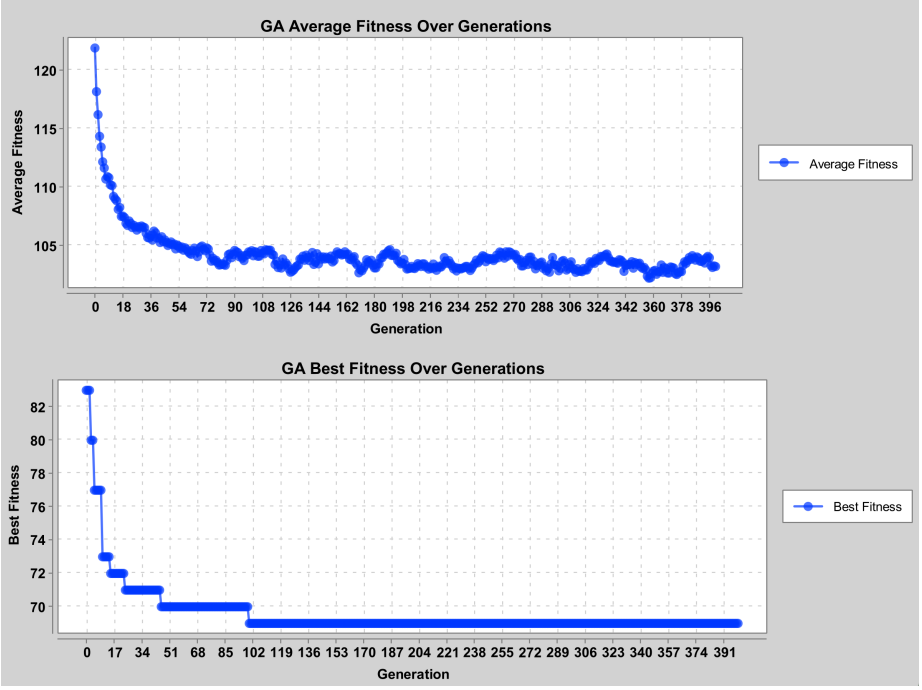


Observe how the ordering of the locations (on the left of the above screen captures) changed to begin at the University of Toronto, then visiting the locations in Niagara from north-most to south-most, resulting in a more efficient and intuitive route from the original randomized order.

2.2 Back-end and GA Development

Progression of the GA has been steady and has seen the group complete a single-TSP solution. Rather than observing the GA from the user's perspective, let us take advantage of the graphing capabilities of the GA, showing the progression of the population to a good solution over iterations. The GA is capable of reading and writing to GeoJson format, facilitating

communication between the front-end and the back-end. The following example of the GA is reading from a GeoJson file containing 15 different locations. Consider the two following screenshots;




```
[Location: 1, Location: 6, Location: 10, Location: 3, Location: 14, Location: 15, Location: 9, Location: 12, Location: 2
[Location: 1, Location: 15, Location: 11, Location: 5, Location: 6, Location: 9, Location: 2, Location: 14, Location: 3,
[Location: 1, Location: 12, Location: 7, Location: 10, Location: 14, Location: 4, Location: 13, Location: 8, Location: 1
[Location: 1, Location: 11, Location: 3, Location: 5, Location: 6, Location: 15, Location: 14, Location: 12, Location: 7
[Location: 1, Location: 3, Location: 11, Location: 15, Location: 5, Location: 6, Location: 10, Location: 4, Location: 13
[Location: 1, Location: 4, Location: 12, Location: 5, Location: 6, Location: 14, Location: 9, Location: 7, Location: 3,
[Location: 1, Location: 15, Location: 9, Location: 12, Location: 10, Location: 2, Location: 4, Location: 14, Location: 3
[Location: 1, Location: 5, Location: 6, Location: 9, Location: 4, Location: 15, Location: 10, Location: 14, Location: 3,
[Location: 1, Location: 2, Location: 5, Location: 6, Location: 9, Location: 13, Location: 12, Location: 4, Location: 14,
[Location: 1, Location: 6, Location: 14, Location: 9, Location: 15, Location: 2, Location: 10, Location: 7, Location: 11
[Location: 1, Location: 4, Location: 2, Location: 15, Location: 11, Location: 9, Location: 14, Location: 13, Location: 3
[Location: 1, Location: 12, Location: 4, Location: 2, Location: 10, Location: 15, Location: 11, Location: 9, Location: 1
[Location: 1, Location: 14, Location: 12, Location: 7, Location: 15, Location: 3, Location: 5, Location: 2, Location: 4,
```

The first screenshot highlights the improvements that the GA makes overtime, particularly by showing how both the average population fitness and the fitness of the best individual improve until essentially plateauing. The next screen capture is simply a random set of individuals from the final population. We have included this to show two main attributes of the algorithm, namely population diversity as well as the fact that the GA's functions have been tuned to ensure that the starting location (ID 1) remains the beginning of a route even after performing crossovers and mutations. Further, as mentioned before, the GA is now capable of writing the best individual's route into GeoJson format, allowing for the interaction between subsystems by having a

standardized data type.

3 Software Engineering Process

Development is in full effect in terms of our project and since we are taking an agile development approach, we are able to adapt to change. Below, we provide a timeline of development in terms of sprints and highlight agile methods used during development.

3.1 Sprints

Sprints have played a valuable role in outlining the work to be completed by group members. We summarize the completed sprints as follows;

- Sprint one's purpose was to see the team complete tasks to begin development of the system such as the implementation of the single-TSP GA, user account creation/management and adding fields to the UI pertaining to features that are to be implemented.
- Sprint two saw the group both refactor and continue the development of the created GA, and refine the work completed on the database side of development. Further, the system was containerized so that one can in fact run the system locally, and this was used to get the screenshots within section 2. Also, Google APIs were used to allow for users to login with their Google credentials and auto-complete location searches.

3.2 Concepts Applied in Development

Concepts from the introductory SW engineering course have been applied within the development of our system. While many agile development methods have been used, the main concepts applied include code review/refactoring, and scrum meetings.

Code review was/is an essential part of the development of the GA and the front-end of the system, as both are large tasks/subsystems. The refactoring of the GA improved overall efficiency of the system (an outlined non-functional requirement) as well as improved code readability and abstraction. Further, members took time to improve upon the code written by other regarding many aspects of front-end of the system including the progression of route visualization and the communication between the front-end and back-end subsystems.

All members have been regularly attending our semi-weekly group meetings (with exception of a few meetings). Such meetings have proved to be valuable in the progression of our system, as members have all been expressing their progress on development, setting deadlines for work to be completed in front

of the group and discussing the changes to be made as well as the next steps in development.

4 Challenges During Development

While numerous smaller challenges have arose during the development of the system, we name the main ones;

- The completion of all defined requirements while maintaining quality work on the satisfied requirements.
- Adhering to development methods, particularly, documenting the completed testing in full (for units and components).
- Time estimation and plan not being accurate of what the realistic capabilities of the team.
- Lack of overall knowledge sharing. The team has been structured in a manner that plays to the strengths of each member, however, what was apparent during the development of this document is that members understand their part very well, while at times having little understanding of the work completed by others in the team.

5 Conclusion and Next Steps

Collectively, we do believe that the development of the system and the overall goal of having a useful system is on track. Despite this, there are aspects of the project that we can make conclusions about, whether that be regarding project velocity, the current product vs the planned product, and using other concepts of software engineering.

Project Velocity
Overall, work is consistently getting completed, and as it stands, no group members have had issues with the work that others have completed.

Current Vs Planned Project
When writing this document, we took time to compare the work that we've completed thus far to the planned sprints, and while the overall velocity of the project is steady, we can't ignore the fact that according to our initial plan, we are behind schedule. Our initial plan had us already completing a multi-TSP solution, as well as route visualization, route assignment for enterprises, and looking into the implementation of real-time updates for routes. While we do not want to exclude these features from the final project, being realistic about our current situation is also beneficial as it will allow more time to enhance the current product and to devote more time to the implementation of additional features with higher priority (like route visualization and a

multi-TSP solution).

Future Sprints and Plans
Scaling back the overall project will not halt work, rather it will allow for more time to complete another necessary aspects of a system (as stated above). Having a system that satisfies attributes such as reliability, security, availability, and maintainability is valuable. As a group, we plan to take time to redefine the the systems requirements as well as the plans for the remaining sprints.

Making the System more Consistent
As seen within this document, at times there are inconsistencies between the capabilities of the GA and the UI. While this coincides with the redefinition of requirements, we understand that the final system should be a "unified" one. For example, the route displayed keeps the specified starting location (has seen within section 2) or the GA is capable of applying the filters that users specify (such as avoiding highways and vehicle type).

6 Group Member Contributions

Below is the work completed by each respective member (Note that the description of the work completed by each member is given by the member themselves);

6.1 Arana Charlebois, Tristan

- Implemented a MongoDB database hosted on Atlas to store user account information, and eventually routes generated by our GA.
- Registration and login flow prototype using Next.js was developed, implementing NextAuth.js as middleware to secure routes and store user data safely.
- Third-party OAuth providers like Google were implemented to allow users to login with already existing accounts, and a password reset flow via nodemailer was made to allow users to reset their password by following a link sent to their email.

6.2 Carvalho, Cam

- Set up a VPS to handle our deployments and built a route calculation system that can scale with user growth using RabbitMQ.
- Created our main landing page to display description of services to users, plus a user dashboard where people can access their data and use the core features within nextjs.

- Created initial project backlog and initial sprints on Trello.

6.3 Corbett, Cole

- Researched and identified required libraries and data for our system and created the corresponding Gradle dependencies file as well as a template so other group members can use the identified libraries.
- Refactored the GA for readability and to better align with a standard GA implementation.
- Added elitism to the GA.
- Added graphing for visualization of the overall fitness of the population and testing purposes to the GA.

6.4 Kassie, Turner

- Authored both initial draft and final submission of all documents required in the course (project proposal, release planning document and progress report one).
- Developed the initial GA used for solving single-TSP, that is the implementation of all core methods in a standard GA.
- Added reading from and writing to standard GeoJson file format allowing the GA to pass information to and from other components within the system.

6.5 Pauls, Andrew

- Created initial project backlog and initial sprints on Trello.
- Ensured that for the first two sprints, the team members were able to fulfill their tasks by asking team members about their task status, tasks were either added or altered with reason.
- Kept meeting minutes and coordinated the meeting space for each meeting.

6.6 Wallace, Jordan

- Created the logo for the product in Illustrator and Photoshop.
- Utilized Google Maps API to allow users to pick auto-completed addresses within Canada when searching a particular location in app.
- Assigned a marker to to given addresses/locations so the user has a visual representation of their route.

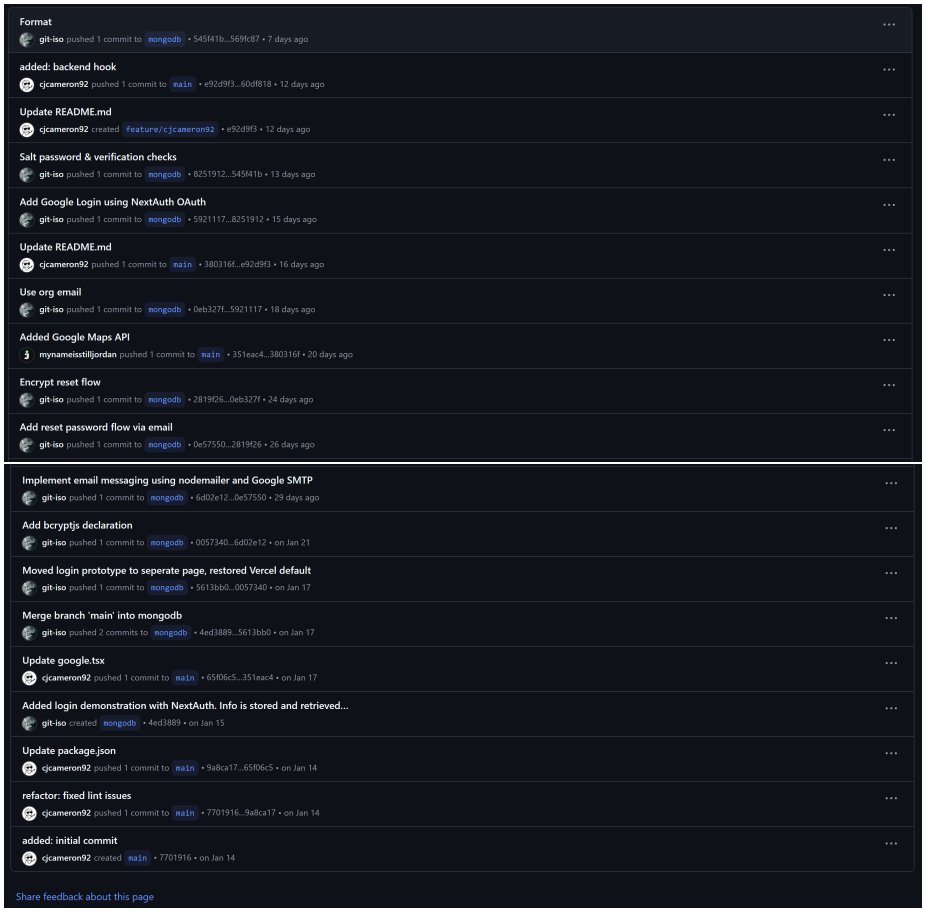
6.7 Work Completed by all Members

- All members brought 5 user stories as well as their respective tasks to be reviewed and combined into the Trello board.
- All members have been actively attending meetings (either in person or virtually) with the exception of a few one-off cases.

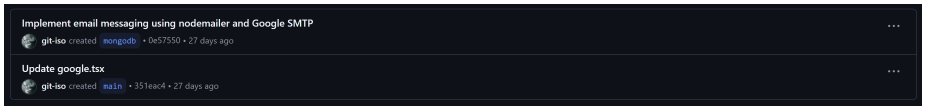
7 Github Logs

The Github logs for each repository within the our Logistics System organization are within their respective subsections as well as a repository used by Arana Charlebois, Tristan for frontend development. The screenshots recorded were taken on February 20th, 2025 for the readers reference.

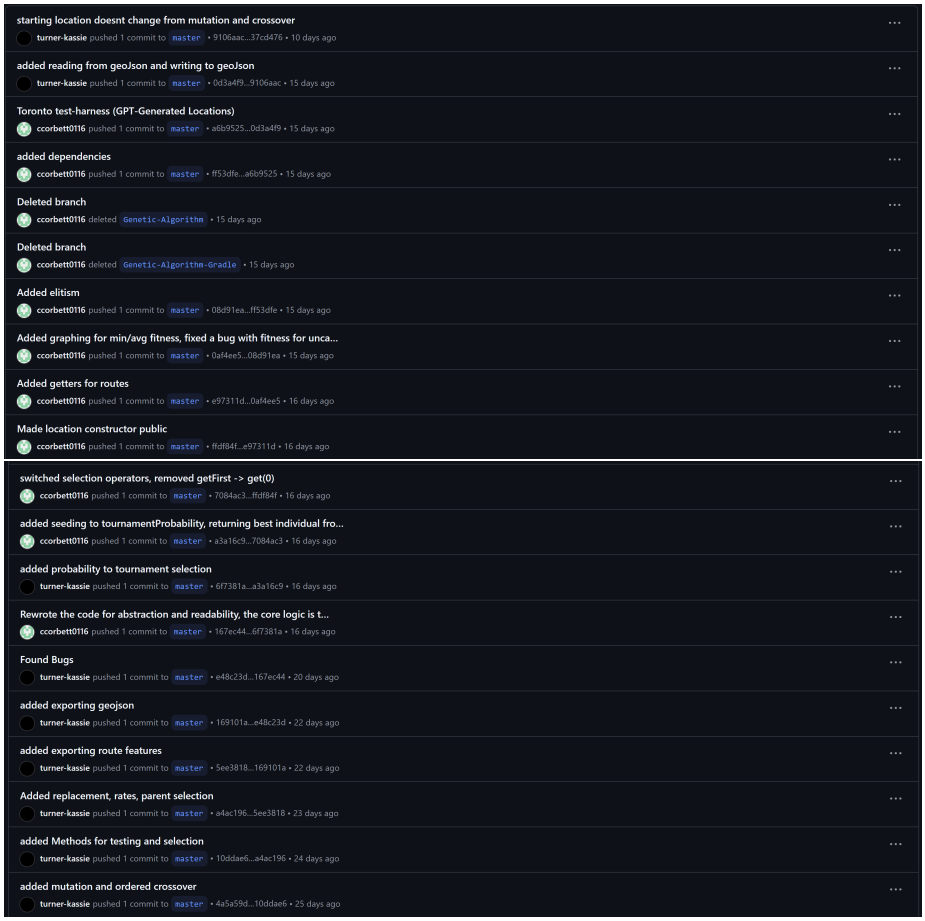
7.1 Front-end Repository

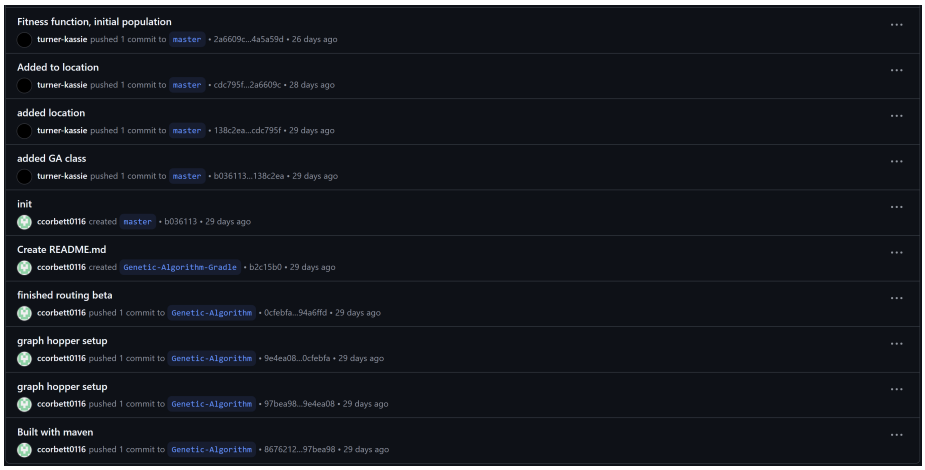


7.2 Front-end ISO Repository (Arana Charlebois, Tristan's repository)



7.3 Logistics System Repository





7.4 RabbitMQ Repository